Multi-agent based modeling of an asset's price

Norman El Fartass

February 2025

Abstract

This document provides a simulation based study of a discrete probabilistic multi-agents complex systems. The objective is to model the price of an asset by assuming that a certain number N of agents, each adopting different trading strategies, interact with each other. To achieve this, we base our approach on an agent interaction model inspired by the study of ants presented in 1993 by Alan Kirman and trading strategies presented by J. Doyne Farmer and Shareen Joshi in 2002.

Contents

1 Introduction

1.1 Model description

The objective is to take advantage of [2] in which Alan Kirman modeled the behavior of ants by using it to simulate the price evolution of a financial asset. To achieve this, we will use concepts introduced by J. Doyne Farmer and Shareen Joshi in [1]. We first have to assume that ant behavior can be transposed to human decision-making.

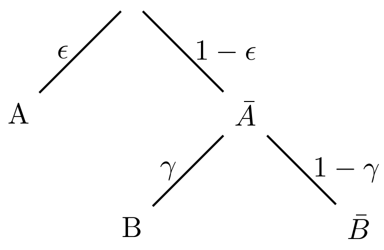
Kirman's model on ant behavior provides an explanation through an agent-based model of a phenomenon observed by entomologists. When two identical food sources are made available to a group of ants, we can observe that they tend to concentrate mainly on one of the two sources instead of being equally distributed. Despite the fact that this phenomenon is not well understood in the case of ants, we can use the model to describe the distribution of traders into two groups, each representing a different trading strategy. The model operates as follows with two classes of agents: trend followers and value investors.

Definition: Trend followers assume that the system has inertia and define their strategy as follows: if the price increases, they buy (which contributes to driving the price up), and if the price decreases, they sell (which contributes to driving the price down). We can assume, to simplify the model, that this class of traders drives the price up if they are the majority and drives it down if they are not. This class of agents will be denoted as **T**.

Definition: Value investors assume that the price should be at a target value and will buy if the asset is undervalued and sell if the asset is overvalued. This will, in the same way as the trend follower, lead to an increase or decrease in the asset's price. This class of agents will be denoted as **V**.

The N agents in the system are randomly distributed into k individuals from **T** and N-k individuals from **V**, where k is drawn uniformly between 0 and N at the beginning.

At each iteration of the model (which can be compared to a time step of one day), two agents are randomly selected from the set of traders. The first agent can change strategy and adopt the opposite strategy by himself with a probability ϵ, or be influenced to change by the other trader with a probability $1-\gamma$. There is no bias in choosing which agent changes, as both are randomly selected. Another pair of agents is then drawn, and this process is repeated M times.



Here, A represents the first agent changing his strategy by himself and B the agent changing his strategy because of the second agent.

At each end of iteration, a new asset price is calculated using the proportion of **T** individuals and **V** individuals as follows:

$$P_{n+1} = P_n + E_{\text{Trend}} + E_{\text{Value}} + Noise$$

Where P_{n+1} is the new price we want to compute, E_{Trend} is the contribution by the **T** agents and E_{Value} the contribution by the **V** agents given by:

$$E_{\text{trend}} = \lambda_T * (2\frac{k}{N} - 1) * P_n$$
$$E_{\text{value}} = \lambda_V * (P_{\text{Presumed}} - P_n) * \frac{N-k}{N}$$

The terms λ_T and λ_V are used to control the strength of each contribution and P_{Presumed} represents the value that **V** agents think the asset should be worth.

1.2 Computational simulation

We can now simulate our model in Python to observe the evolution of an asset's price. For this example, we have chosen Bitcoin, as the hype around its price on social media makes the assumption that agents influence each other more "realistic".

```python
import numpy as np
import matplotlib.pyplot as plt


N = 500  # Number of agents (investors)
M = N//2  # Number of interactions for each iteration
epsilon = 0.01  # Probability that an agent changes his
    strategy by himself
delta = 0.9  # Probability that an agent does not change
    strategy during an interaction (1-delta represents the
    probability that it changes)
steps = 1000  # Number of iterations
initial_price = 30000  # Initial price
target_price = 50000  # Target price
lambdaT = 0.05  # Influence of trend followers on the price
lambdaV = 0.02  # Influence of value investors on the price
noise = 0.02  # Intensity of random noise


#We first build the array containing each agents
state = np.random.choice([0, 1], size=N)  # 0: Value Investor
    , 1: Trend Follower
#This implementation is helpful as we can just add all the
    values to find k
price_list = [initial_price]  # Price historic
price = initial_price

for _ in range(steps):
    # Agents interactions
    for _ in range(M):
        i, j = np.random.randint(N, size=2)  # We select two random agents
        if np.random.rand() < epsilon:
            state[i] = 1 - state[i]  # The first agent changes by itself
        elif np.random.rand() < (1 - delta):
            state[i] = state[j]  # The second agent makes the first one change his strategy


    k = np.sum(state)

    # Price update
    Etrend = lambdaT * (2 * k/N - 1) * price
    Evalue = lambdaV * (target_price - price) * (N-k)/N
    noise_effect = np.random.randn() * noise * price # It makes the graph feel more realsistic (otherwise it looks like a continuous curve...)

    price += Etrend + Evalue + noise_effect
    price = max(1000, price)  # Prevents the price to be negative by setting a minimum value (can’t take 0 otherwise it might stop the simulation)
    price_list.append(price)




plt.figure(figsize=(10, 5))
plt.plot(price_list, label = f"Simulated Bitcoin price with: \n N = {N} \n M = {M} \n epsilon = {epsilon} \n delta = {delta}")
plt.axhline(50000, color="red", linestyle="--", label="Target price of value investors")
plt.xlabel("Iterations")
plt.ylabel("Bitcoin price")
plt.legend()
plt.title(f"Bitcoin price simulation during {steps} days")
plt.show()
```

Figure 1: Model simulation with Python

The initialization variables at the beginning are set as an example of how we could model the price and interactions. We can now see how the first simulation of the model looks with these inital values:

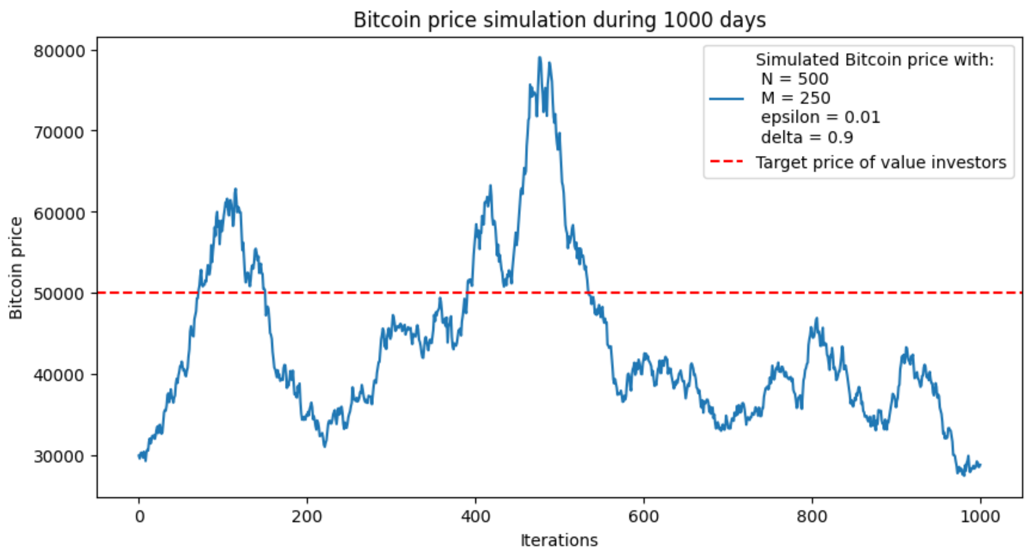


Figure 2: Bitcoin price simulation with the model

As we can see, the result is not far from what we would have expected for a Bitcoin price chart with several phases of increases and decreases.

2 Analysis of the model's properties

2.1 Simulation based analysis

We can use intuition and the way the model was designed to deduce many consequences regarding price evolution with agent interactions. First, we can observe that the number of agents has a significant role in the simulation. In a small group of agents, it is very likely that a large proportion of individuals belong to one group compared to the other. This can lead to an unpredictable evolution of the price. For a group of around 50 individuals, we can get these results :

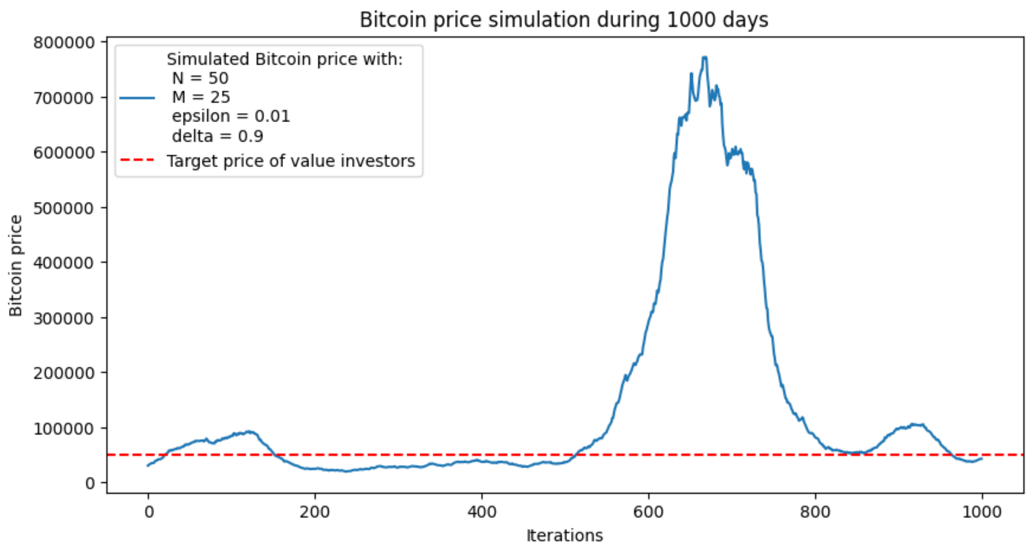


Figure 3: Bitcoin price simulation with few agents

We have here an abrupt spike that decreases as fast as it increased. Although it might happen in some rare cases with an asset price, it is not very natural and quite unexpected.

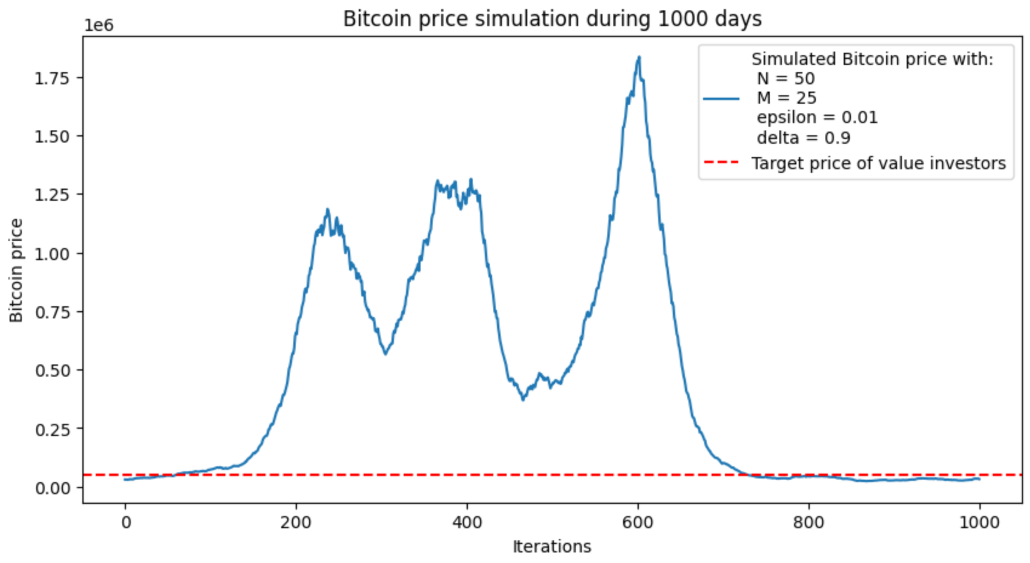


Figure 4: Bitcoin price simulation with few agents

We can see here that the price goes way above what we can consider being normal for this period of time. It is also due to the number of agents being low.

On the other hand, a high value of N will lead to a more accurate modelisation of the price as we can expect in probability theory with the law of large number.

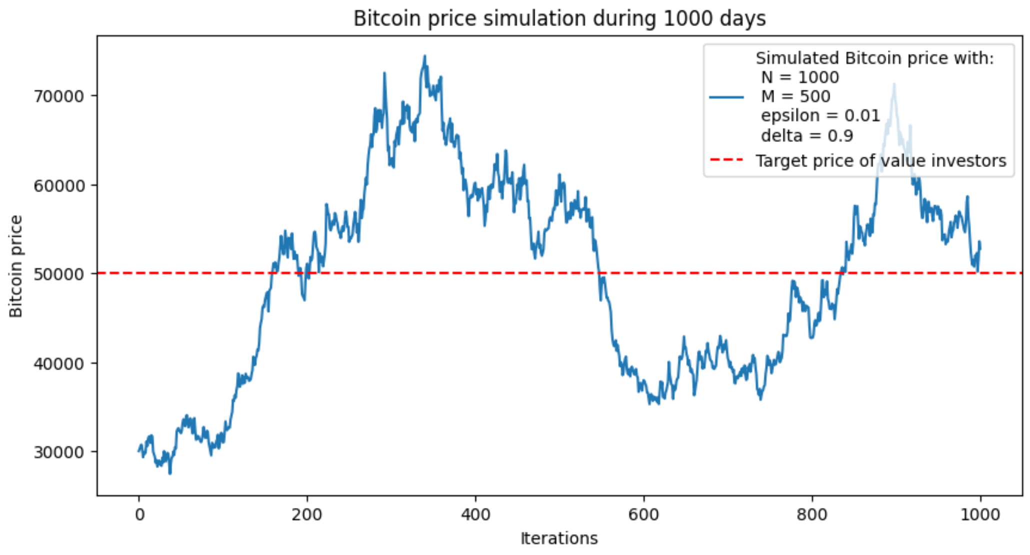


Figure 5: Bitcoin price simulation with 1000 agents

We can see in this figure the bubble and crisis phenomena that the model is trying to replicate. Other variables can also influence the price value and its evolution. The variable M, for example, defines the number of interactions that occur at each iteration (i.e., each day). The higher this variable is (e.g., M=N), the faster the trends and cycles will be. Conversely, a lower value of M (e.g., M=N//4) would slow down the emergence and length of the cycles.

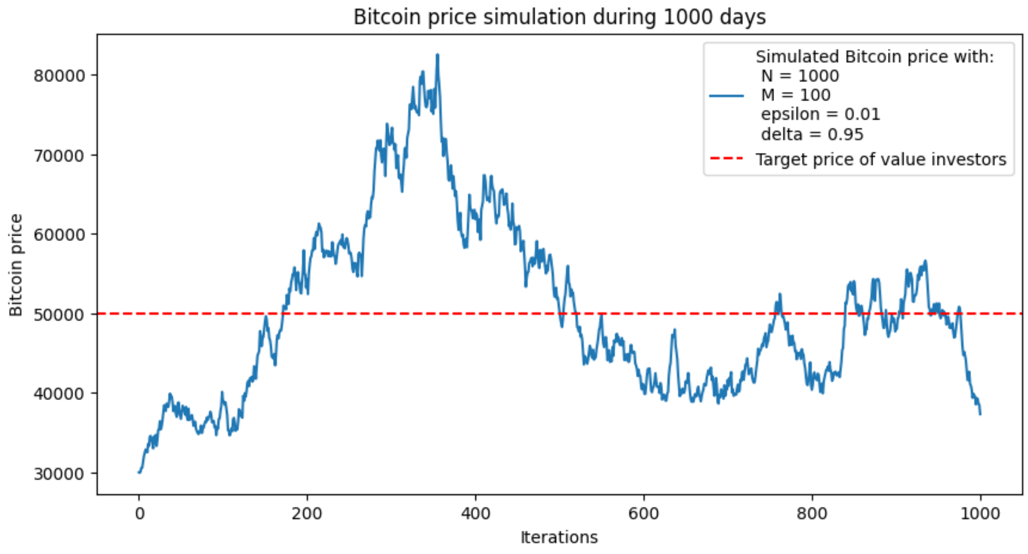


Figure 6: Extended price cycle over a 1000 agents simulation

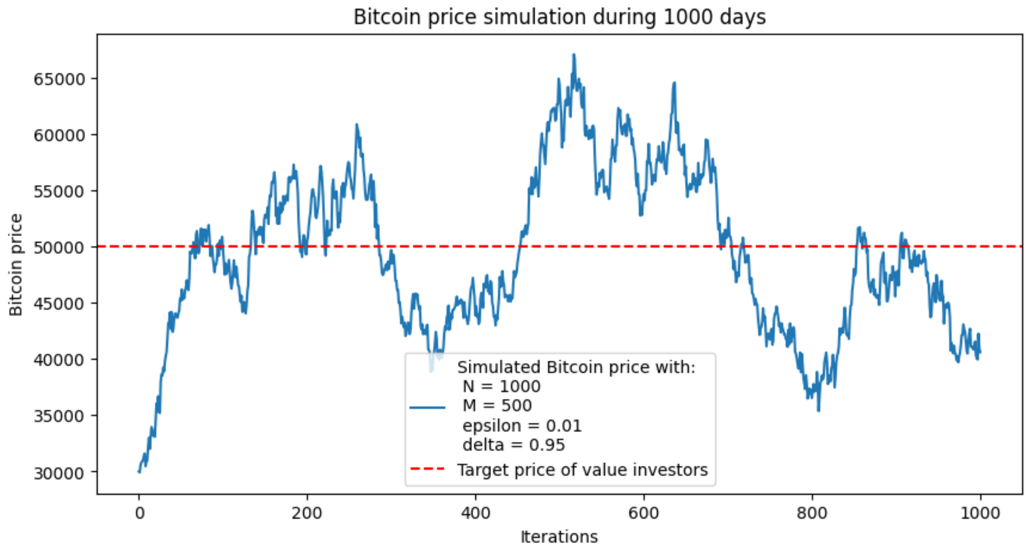


Figure 7: Short price cycles over a 1000 agents simulation

Finally, the parameter δ influences the duration of trends and cycles, as it affects the frequency at which individuals change strategies. The higher δ is, the slower investors will change their strategy so it will result in long trends.

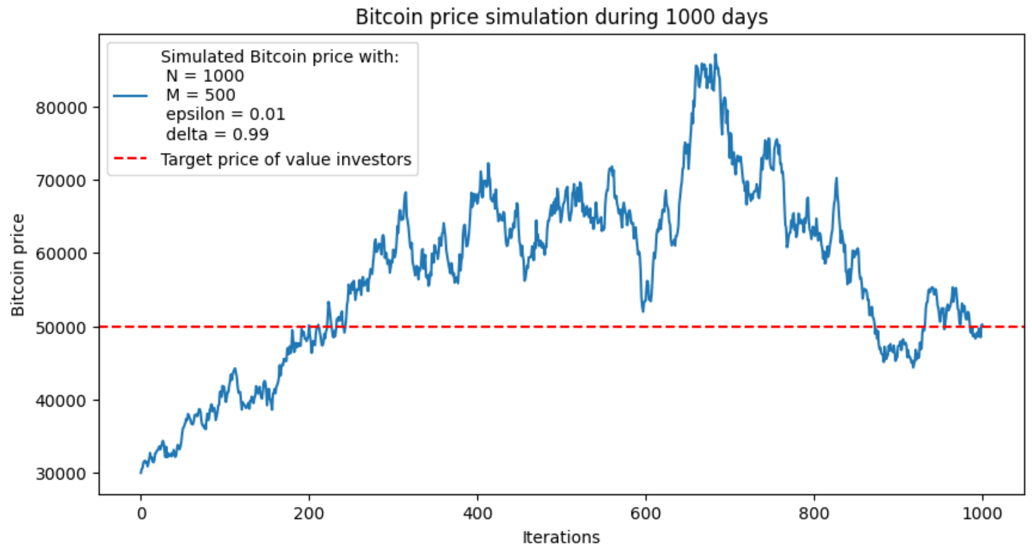


Figure 8: Simulation with a high δ value

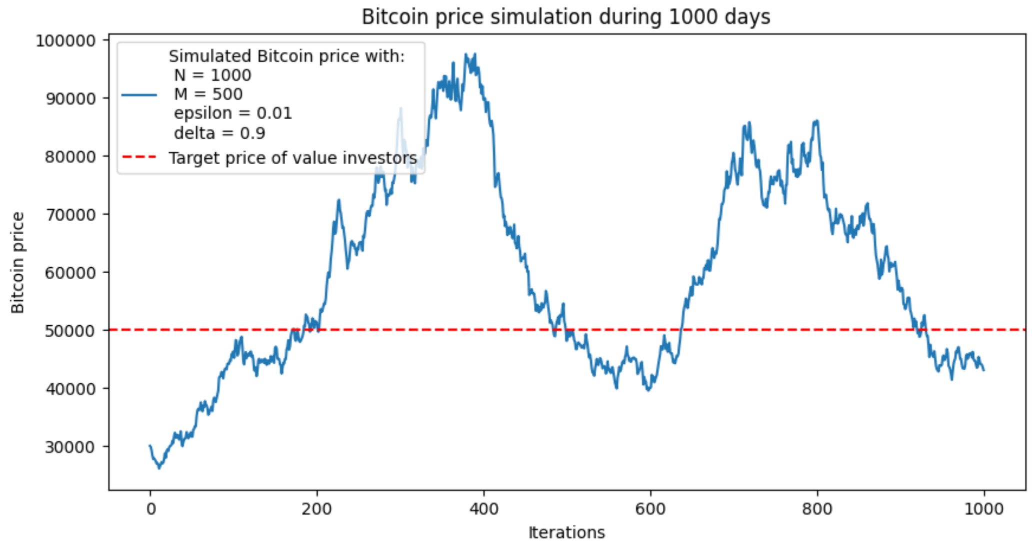


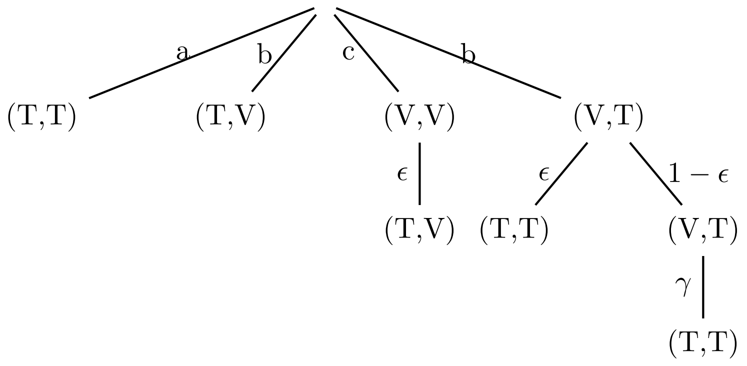
Figure 9: Simulation with a low δ value

We can observe that the model will always be cyclical and that the price will never converge to a precise value. This is due to the fact that at each iteration, there will always be probabilistic interactions between agents, creating a new dynamic. Moreover, if the price is too high, value investors will apply a strong correction, leading to a decrease in value. If this value drops below the target value, the opposite phenomenon will occur.

2.2 Mathematical approach

We can use mathematics to try to predict or understand the evolution of the distribution of agents between the two groups. The model operates in such a way that we only need to know the result at iteration n to determine the result at iteration $n + 1$. Therefore, it is Markovian, and we can analyze the probabilities governing the evolution of this stochastic process easily. Once we know how k is moving, it is easy to deduce the probabilities on the price of the asset. We will only focus on the case where M = 1 to simplify the calculations. We will also consider that $1 - \delta = \gamma - \gamma\epsilon$ which will help represent the path on a tree where δ is the parameter we used in the simulation.

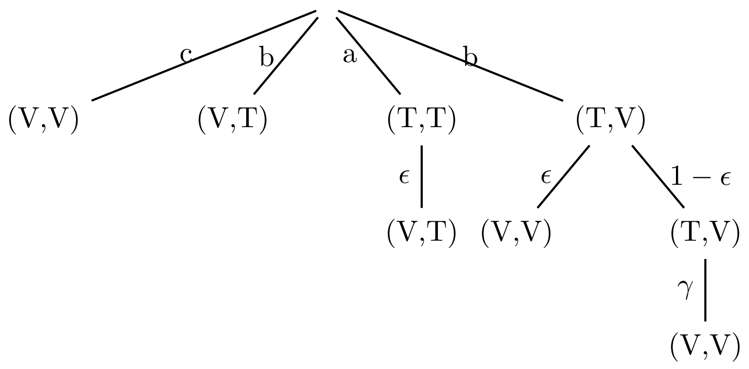
We first want to know $P(k \to k + 1)$. At first, two agents are drawn from the set randomly. We have four possible combination each with a different probability to occur: (T,T), (T,V), (V,V) and (V,T). If we want the case where $k \to k + 1$, we can't consider (T,T) and (T,V) as the first agent is already one of the k trend followers. We can model the situation with this tree:



With a = $\frac{k^2}{N^2}$, b = $\frac{k(N-k)}{N^2}$, c = $\frac{(N-k)^2}{N^2}$ and $a + 2b + c = 1$

We can conclude that $P(k \to k + 1) = \frac{(N-k)^2}{N^2}\epsilon + \frac{k(N-k)}{N^2}(\epsilon + (1 - \epsilon)\gamma)$

Using the same reasoning, we can compute $P(k \to k - 1)$.



We then have $P(k \to k-1) = \frac{k^2}{N^2}\epsilon + \frac{k(N-k)}{N^2}(\epsilon + (1-\epsilon)\gamma$

The markov chain is represented by these formulas:

$$\begin{aligned}
P(k \to k+1) &= \frac{N-k}{N}[\epsilon + \frac{k}{N}(1-\epsilon)\gamma] && (1)\\
P(k \to k-1) &= \frac{k}{N}[\epsilon + \frac{N-k}{N}(1-\epsilon)\gamma] && (2)\\
P(k \to k) &= 1 - P(k \to k+1) - P(k \to k-1) && (3)\\
&= 1 - \epsilon - 2\frac{k(N-k)}{N^2}(1-\epsilon)\gamma
\end{aligned}$$

3 Conclusion

In this work, we explored a discrete probabilistic multi-agent system applied to financial asset price modeling inspired by Alan Kirman's work [2] and J. Doyne Farmer and Shareen Joshi's research on trading strategies [1].

By categorizing agents into trend followers and value investors, the model replicates key financial phenomena, such as market bubbles and cyclical price movements. The stochastic nature of agent interactions and the design of the model ensures that prices never converge to a fixed equilibrium. We instead notice fluctuations and cycles.

The computational simulations highlight the sensitivity of price dynamics to parameters such as the number of agents (N), interaction frequency (M), and switching probabilities (ϵ, γ, δ).

An improvement could be the consideration of exogenous shocks, such as the announcement of a future crisis or political instability. We can also add various differents strategies and model interactions according to social studies. Finally, it is important to remain critical of multi-agent models because, although they can provide a description of reality, this description will never account for all the factors that can influence the phenomenon being modeled. Additionally, assumptions must be made about human behavior, which in reality is difficult to predict.

References

[1] Farmer, J. Doyne, and Shareen Joshi. *The Price Dynamics of Common Trading Strategies*. Journal of Economic Behavior & Organization, vol. 49, no. 2, October 2002, pp. 149-71. DOI.org (Crossref), https://doi.org/10.1016/S0167-2681(02)00065-3.

[2] Kirman, A. *Ants, Rationality, and Recruitment*. The Quarterly Journal of Economics, vol. 108, no. 1, February 1993, pp. 137-56. DOI.org (Crossref), https://doi.org/10.2307/2118498.